IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No.	10/699,629
Filing Date	October 30, 2003
Inventorship	Burdick, et al.
Applicant	Microsoft Corporation
Group Art Unit	2116
Examiner	Brown, M.
Attorney's Docket No.	MS1-1654US

Title: Configuration Settings

To: The Honorable Commissioner for Patents
Mail Stop Appeal Brief- Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: William J. Breen III (Tel. 509.755.7253; Fax 509.755.7252)
Customer No. 22971

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 C.F.R. § 1.192. The Filing Fee, as set forth in 37 C.F.R. § 1.17(c), is submitted herewith.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, by way of assignment from Burdick et al., who is the named inventive entity and is captioned in the present brief.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

Claims 1-47 are pending and are the subject of this appeal.

STATUS OF AMENDMENTS

None.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Configuration settings are described which may be utilized to indicate a configuration of an application. For example, a remotely based application may be executed on a content server to provide content, such as a movie, web page, and so on, to a client-based application that is executed on a set-top box to view the content. To enable the remotely based application to work with the client-based application, it is desirable that the remotely based application have configuration settings that are valid for use with the client-based application, and vice versa. A configuration module is described which allows configuration settings of the applications to be validated, one to another, such that validity of the applications may be determined. *See Application, Paragraph [0006]*.

Also described is a documenter that generates a configuration file that contains the configuration settings of the application. The documenter may write a field and descriptions thereof to document the configuration settings of the application. Each such field, for instance, may have a plurality of descriptions of conditions for the field. Each such description and the corresponding field may provide a configuration setting, such as a value constraint, default value, textual description, and so on. For instance, the field "letter grade" may have a value constraint that specifies that the letters "A", "B", "C", "D", and "F" for a value of the field. Therefore, the field "letter grade" and the value constraint describe a configuration setting of the application. In an exemplary implementation, a method includes executing a documenter to find a plurality of fields in an application. The application is composed of computer instructions that have attributes. Each attribute provides a description of a condition for a respective field. A configuration file is formed having a plurality of configuration settings of the application. Each configuration setting includes one of the fields and the description of the condition for the respective one of the fields. The configuration file is then output. *See Application, Paragraph [0007]*.

In an additional implementation, the configuration file generated by the documenter may be utilized through execution of the configuration module to validate configuration settings. *See Application, Paragraph [0008]*.

Independent Claim 1 recites a method comprising validating a configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) of a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) for use with a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) (e.g., reference number 306, FIG. 3; paragraph [0037]), wherein:

the configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) including a first field and a first description of a first condition for the first field;

the second application (e.g., reference number 124, FIG. 2; paragraph [0023]) is composed of computer instructions, the computer instructions having an attribute, the attribute providing a second description of a second condition for a second field; and

the validating includes:

if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition (e.g., reference number 404, FIG. 4; paragraph [0041]); and

if met, then determining that the configuration setting is valid for use with the second application (e.g., reference number 404, FIG. 4; paragraph [0041]).

Dependent Claim 2 recites a method as described in claim 1, wherein the first and second conditions are value constraints (e.g., paragraphs [0007], [0021]. [0030]) selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent Claim 9 recites a method comprising:

reading a first configuration setting (e.g., reference numbers 206(1)-206(J); paragraph

[0032]) of a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) including a first field and a first description of a first condition for the first field;

examining a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) to find a second configuration setting that corresponds to the first configuration setting, wherein the second application is composed of computer instructions, the computer instructions having an attribute, the attribute providing a second description of a second condition for a second field, the second configuration setting having the second field and the second description (e.g., reference number 506, FIG. 4; paragraph [0052]); and

comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application (e.g., reference number 404, FIG. 4; paragraph [0041]).

Dependent Claim 10 recites a method as described in claim 9, wherein the first and second conditions are value constraints (e.g., paragraphs [0007], [0021]. [0030]) selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent claim 14 recites a method comprising:

executing a documenter (e.g., reference number 230, FIGS. 2 and 5; paragraph [0034]) to find a plurality of fields in an application (e.g., reference number 120, FIG. 1; paragraph [0023]), wherein:

the application (e.g., reference number 120, FIG. 1; paragraph [0023]) is composed of computer instructions;

the computer instructions having attributes; and

each said attribute providing a description of a condition for a respective said field,

forming a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) having a plurality of configuration settings of the application, wherein each said configuration setting includes one said field and the description of the condition for the one said field (e.g., reference number 512, FIG. 5; paragraph [0054]);

outputting the configuration file; and

utilizing the configuration file to validate configuration settings.

Dependent Claim 15 recites a method as described in claim 14, wherein the first and second conditions are value constraints (e.g., paragraphs [0007], [0021], [0030]) selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent claim 19 recites a method comprising:

generating a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) having a plurality of configuration settings (e.g., reference numbers 206(1)-206(J); paragraph [0032]) derived from a first application (e.g., reference number 512, FIG. 5; paragraph [0054]), wherein:

the application is composed of computer instructions;

the computer instructions having attributes;

each said attribute providing a description of a condition for a field; and

each said configuration setting having one said field and a corresponding said description; and

validating whether the first application is valid for use with a second said application (e.g., reference number 124, FIG. 2; paragraph [0023]) (e.g., reference number 206, FIG. 3; paragraph [0037]) by comparing each said configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) of the first application with a corresponding said configuration setting of the second said application (e.g., reference number 124, FIG. 2; paragraph [0023]) to determine whether each said condition of the first application is met by a corresponding said condition of the second application (e.g., reference number 124, FIG. 2; paragraph [0023]) (e.g., reference number 404, FIG. 4; paragraph [0041]).

Dependent Claim 20 recites a method as described in claim 19, wherein the first and second conditions are value constraints (e.g., paragraphs [0007], [0021], [0030]) selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent claim 24 recites a computer-readable medium comprising computer-executable instructions that, when executed by a computer, direct the computer to:

read a first configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) of a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) that includes a first field and a first description of a first condition for the first field; and

validate whether the first condition is met by a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) (e.g., reference number 206, FIG. 3; paragraph [0037]), wherein:

the second application is composed of computer instructions;

the computer instructions have an attribute that provides a second description of a second condition for a second field; and

the first condition is validated through comparison with the second condition (e.g., reference number 404, FIG. 4; paragraph [0041]).

Dependent Claim 25 recites a computer-readable medium as described in claim 24, wherein the first and second conditions are value constraints selected from the group consisting of (e.g., paragraphs [0007], [0021], [0030]):

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent Claim 28 recites a computer comprising:

a processor (e.g., reference number 202, FIG. 2); and

memory (e.g., reference number 204, FIG. 2) configured to maintain:

a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) composed of computer instructions, the computer instructions having an attribute, the attribute providing a first description of a first condition for a first field;

a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) including a configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) of a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) having a second field and a second description of a second condition for the second field; and

a configuration module (e.g., reference numbers 126-132, FIG. 2; paragraphs [0024]-[0025]) that, when executed on the processor, validates the configuration setting for use with the first application (e.g., reference number 206, FIG. 3;

paragraph [0037]) by comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met, then determining that the configuration setting is valid for use with the first application (e.g., reference number 404, FIG. 4; paragraph [0041]).

Independent claim 32 recites a computer comprising:

a processor; and

memory configured to maintain:

a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) composed of computer instructions, the computer instructions having an attribute, the attribute providing a first description of a first condition for a first field, and wherein a first configuration setting includes the first description and the first field;

a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) including a second configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) of a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) having a second field and a second description of a second condition for the second field; and

a configuration module (e.g., reference numbers 126-132, FIG. 2; paragraphs [0024]-[0025]) that, when executed on the processor, validates the first configuration setting for use with the second application (e.g., reference number 206, FIG. 3; paragraph [0037]) by comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition, and if met, then determining that the first configuration setting is valid for use with the second application (e.g., reference number 404, FIG. 4; paragraph [0041]).

Independent claim 36 recites a content server comprising:

a broadcast transmitter configured to provide media content to a client in response to a request from the client;

a processor; and

memory configured to maintain:

a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) that when executed provides media content for broadcast by the broadcast transmitter, wherein the application is composed of computer instructions, the computer instructions have attributes, and each said attribute provides a description of a condition for a field; and

a documenter (e.g., reference number 230, FIGS. 2 and 5; paragraph [0034]) that is executable on the processor to generate a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) having a configuration setting of the first application, wherein the configuration setting includes the field and the description of the condition for the field (e.g., reference number 512, FIG. 5; paragraph [0054]).

Dependent Claim 37 recites a content server as described in claim 36, wherein the condition is a value constraint selected from the group consisting of (e.g., paragraphs [0007], [0021], [0030]):

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent Claim 40 recites a content server comprising:

a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) composed of computer instructions, the computer instructions having attributes, and each said attribute providing a description of a condition for a field;

a configuration module (e.g., reference numbers 126-132, FIG. 2; paragraphs [0024]-[0025]) that is executable to validate whether each said condition is met by a second application (e.g., reference numbers 126-132, FIG. 2; paragraphs [0024]-[0025]); and

a documenter (e.g., reference number 230, FIGS. 2 and 5; paragraph [0034]) that is executable to generate a configuration file (e.g., reference number 304, FIG. 3; paragraph [0037]) having a configuration setting of the first application, wherein the configuration setting includes the field and the description of the condition.

Dependent Claim 42 recites a content server as described in claim 40, wherein each said condition is a value constraint selected from the group consisting of (e.g., paragraphs [0007], [0021], [0030]):

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

Independent Claim 45 recites a system comprising:

a network (e.g., reference number 106, FIG. 1);

a first computer (e.g., reference number 114, FIG. 2) communicatively coupled to the network and including a first application (e.g., reference number 120, FIG. 1; paragraph [0023]) composed of computer instructions, the computer instructions having a first attribute that provides a first description of a first condition for the first field, wherein a first configuration setting (e.g., reference numbers 206(1)-206(J); paragraph [0032]) includes the first field and the first description;

a second computer (e.g., reference number 104, FIG. 2) communicatively coupled to the network and including:

a second application (e.g., reference number 124, FIG. 2; paragraph [0023]) composed of computer instructions having a second attribute that provides a second description of a second condition for a second field; and

a configuration module (e.g., reference numbers 126-132, FIG. 2; paragraphs [0024]-[0025]) that is executable by the second computer to validate the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition (e.g., reference number 404, FIG. 4; paragraph [0041]).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 6-9, 13-14, 18-19, 23-24, 28-30, 32-34, 36 and 40-41 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over U.S. Patent Application Publication No. 2004/0243997 to Mullen et al. (hereinafter "Mullen").

2. Claims 2-4, 10-12, 15-17, 20-22, 25-27, 31, 35, 37-39 and 42-47 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Mullen in view of U.S. Patent Application Publication Number 2002/0026436 to Joory et al. (hereinafter "Joory").

ARGUMENT

First Ground of Rejection Claims 1, 6-9, 13-14, 18-19, 23-24, 28-30, 32-34, 36 and 40-41 satisfy the requirements of 35 U.S.C. § 102(e) and therefore are not anticipated by Mullen.

Claim 1 is directed to a server system. Claim 1 recites in part:

- if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition; and
- if met, then determining that the configuration setting is valid for use with the second application.

The Applicant respectfully submits that the Mullen reference fails to disclose at least these features.

Mullen describes a method in which program components are copied from a first file directory to a second file directory during installation. *See Mullen, Page 1, paragraph [0006]*. The Mullen system only determines the application configuration settings of the first file directory and then copies the setting to the second file directory. *See Mullen, Page 1, paragraph [0009]*. Nowhere does Mullen disclose validating including "if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition" or "if met, then determining that the configuration setting is valid for use with the second application" as recited in claim 1. Mullen simply is a copy functionality which is capable of determining application configuration settings of a first directory and then copying them to a second file directory. *See Mullen, Page 1, paragraph [0009].* Mullen, Paragraph [0029] is incorrectly cited for this disclosure. The entirety of Mullen paragraph [0029] is reproduced below for convenience.

> [0029] In certain implementations, the configuration setting entry 80 (FIG. 3) may indicate a directory location and file or parameter name for the operating system or application program configurations settings 84a, 84b . . . 84n for the preexisting installation to copy and the target directory location of where to write the copied preexisting configuration settings 106, 110. The configuration settings 84a, 84b . . . 84n may indicate the source of the configuration settings to copy as relative to the root of the preexisting directory locations 102 and the target location to write the configuration settings as relative to the root of the directory locations 122 for the new installation. Alternatively, the configuration settings 84a, 84b . . . 84n may indicate the source and target locations for the configuration settings to copy as absolute file directory locations. If the configuration settings to copy are maintained in a file, then the file would be copied over. If the configuration settings to copy comprise parameters in a data structure, such as a registry file or other file, then the installation program 52 would open that data structure to access the configuration settings to copy, and then open the corresponding data structure in the new installation to update with the copied configuration setting value. *See Mullen, Paragraph [0029].*

As shown in the above paragraph, Mullen discloses determining and copy functionality. Paragraph [0029] does not "compare the first description of the first condition with the second description of the second condition."

Claim 1 recites a method of validating a configuration setting of a first application including in part "if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition." Neither paragraph [0029], nor anywhere does Mullen teach the recited portion. Additionally, Mullen fails to teach if met, "then determining that the configuration setting is valid for use with the second application." Mullen does not disclose this capability, as the Mullen reference merely determines "[i]f the configuration settings to copy comprise parameters in a data structure . . . then the installation program 52 would open that data structure to access the configuration settings to copy . . ." and subsequently copies the data to the second file directory (the new installation).

The Examiner, in the *Response to Arguments*, asserted that "Mullen discloses that if an entry 90 is included in the application configuration list 56, then comparing a location relative to the root location of the currently installed operating system of where the configuration setting to copy and the target location of where to write the copied configuration setting are located relative to the root of the operating system or application installed from the system installation (see paragraph 023, lines 10-17). This will validate whether the entry 80 is included in the configuration list 56." The asserted portion of Mullen is excerpted as follows for the sake of convenience:

> [0023] FIG. 3 illustrates information maintained in the application/OS configuration list 56. The application/OS configuration list 56 may include entries 80, where each entry identifies an operating system or application program 82

> included in the installation and one or more configuration settings 84a, 84b . . . 84n that are to be copied from the configuration settings of the currently installed operating system or application to the configuration settings used by the operating system and applications installed from the system installation package 50. The configuration setting 84a, 84b . . . 84n information would indicate a location relative to the root location of the currently installed operating system where the configuration setting to copy and the target location of where to write the copied configuration setting are located relative to the root of the operating system or application installed from the system installation package 50. As discussed, the copied configuration setting may comprise a file or a parameter or value accessed from a file. The configuration setting information may be implemented in different data formats and data structures than shown in FIG. 3. *See Mullen, Paragraph [0023].*

As is readily apparent from the above asserted section, however, copying of configuration setting is again disclosed, with no mention of a condition for a field nor comparison of these conditions. Accordingly, a *prima facie* case of anticipation has not been established, and withdrawal of the rejection is respectfully requested.

Claim 5 is allowable based on its dependency from Claim 1. Claim 5 is further allowable as Mullen fails to teach an attribute being "a declarative tag." Mullen paragraph [0027] cited for this proposition only discloses logic implemented during installation. *See Mullen, Pages 2-3, paragraph [0027].* In the same passage, Mullen only discloses 1) the operating system (OS) may continue to run, 2) the creation of a new root directory (for the "new installation"), and 3) rebooting will cause the new installed OS to run. *See Mullen, Pages 2-3, paragraph [0027].* Nowhere else does

Mullen correct the deficiency of teach the features recited in Claim 5. Removal of the pending rejection is respectfully requested.

Claims 6, 7 and 8 are allowable based on the same rational as Claim 1 from which the instant claims depend. Furthermore, with respect to Claim 6, as Mullen fails to "validate" as recited in Claim 1, Mullen is incapable of "communicating a result" of the validation as recited in Claim 6. Mullen, paragraph [0029] does not correct this defect. Removal of the pending rejection is respectfully requested.

Claim 9 is directed to a method and recites in part:

- examining a second application to find a second configuration setting that corresponds to the first configuration setting, wherein the second application is composed of computer instructions, the computer instructions having an attribute, the attribute providing a second description of a second condition for a second field, the second configuration setting having the second field and the second description; and
- comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application.

The Mullen reference fails to disclose at least these features.

Mullen FIG. 4 items 128 and 130 are cited as teaching "examining a second application . . ." This is incorrect. The Mullen reference does not examine a second application. Rather, Mullen simply teaches a copy functionality which does not include examination of a second application. *See Mullen, Page 3, paragraph [0029].* Mullen is only concerned with copying a first file directory to generate an image for a second directory. Mullen ends with this copy functionality and does not compare "the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application" as recited in Claim 9. Not only does Mullen fail to disclose the foregoing passage, but Mullen would not require this functionality as Mullen is limited to copying to generate an operating system image, such as when installing an operating system. *See Mullen, Abstract.*

It is respectfully submitted that the reference to Mullen paragraph [0029] is incorrect as this passage (reproduced above) does not disclose "comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application." Mullen Paragraph [0029] only teaches "[i]f the configuration settings to copy comprise parameters in a data structure. . . then the installation program 52 would open that data structure to access the configuration settings to copy. . ." and subsequently copies the data to the second file directory (the new installation). Nowhere is comparing disclosed or even suggested. Removal of the pending rejection is respectfully requested.

Claim 13 is allowable based on its dependency from Claim 9. As a *prima facie* case of anticipation has not been shown with respect to Claim 9, removal of the pending rejection is respectfully requested.

Claim 14 has been amended it is believed to be in a condition for allowance which is earnestly solicited. Nowhere in the asserted references is a method including utilizing the configuration file to validate configuration settings disclosed or suggested. Removal of the pending rejection is respectfully requested.

Claim 18 is allowable as Claim 18 recites one or more computer readable media . . . that perform the method as recited in claim 14. Mullen fails to teach or disclose computer readable media having this performance capability. Removal of the pending rejection is respectfully requested.

Claim 19 is allowable as nowhere does Mullen disclose or teach a method including "validating whether the first application is valid for use with a second said application by comparing each said configuration setting of the first application with a corresponding said configuration setting of the second said application to determine whether each said condition of the first application is met by a corresponding said condition of the second application." Mullen fails to teach any validation as Mullen merely provides a copy functionality. Removal of the pending rejection is respectfully requested.

Claim 23 is allowable as it performs the method of Claim 19. Claim 19 is allowable as nowhere does Mullen teach "validating whether the first application is valid for use with a second said application by comparing each said configuration setting of the first application with a corresponding said configuration setting of the second said application to determine whether each said condition of the first application is met by a corresponding said condition of the second application." As Claim 19 is allowable based on the foregoing rational, Claim 23 is allowable as the Mullen reference fails to disclose the features of Claim 19, nor does Mullen disclose "[o]ne or more computer-readable media comprising computer executable instructions that, when executed, perform the method as recited in claim 19." Removal of the pending rejection to Claims 23 is respectfully requested and allowance solicited.

Claim 24 is pending a rejection under 35 U.S.C. 102(e) over Mullen. Applicants disagree. Claim 24 in part recites:

- validate whether the first condition is met by a second application, wherein:
  the second application is composed of computer instructions;
  the computer instructions have an attribute that provides a second description of a second condition for a second field; and
  the first condition is validated through comparison with the second condition.

Mullen fails to teach computer-readable medium which when executed causes a computer to "validate whether the first condition is met by a second application . . ." Mullen item "128" of FIG. 4 was incorrectly cited for this teaching. FIG. 4 item 128 is the copied "Installed Application(s)". At Mullen paragraph [0028] (the relevant portion of the reference) describes accessing the one or more application images and copying the images to a location in the created directory location to provide the installed application(s) 128. *See Mullen, Page 3, paragraph [0028].* At no time does the Mullen reference validate, as item 128 is a copy of the application images. Removal of the pending rejection is respectfully requested and allowance is solicited.

Claim 28 currently is rejected. Applicants disagree with the rejection. Claim 28 recites a computer, in part, including memory. The memory configured to maintain: "a configuration file including a configuration setting of a second application having a second field and a second description of a second condition for the second field;" and "a configuration module that, when executed on the processor, validates the configuration setting for use with the first application by comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met, then determining that the configuration setting is valid for use with the first application." The Mullen file system 100 does neither. The Mullen system fails to include a configuration module that . . . "validates the configuration setting for use with the first application by comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met, then determining that the configuration setting is valid for use with the first application." While Mullen paragraph [0029] is cited for this proposition the passage does not recite this asserted teaching. No other portion of Mullen corrects this defect. Paragraph [0029] teaches a copy functionality which does not validate. Instead, Mullen discloses, "[i]f the configuration settings to copy comprise parameters in a data structure, such as a registry file or other file, then the installation program 52 would open that data structure to access the configuration settings to copy, and then open the corresponding data structure in the new installation to update with the copied configuration setting value." *See Mullen, Page 3, paragraph [0029].* Removal of the pending rejection is respectfully requested and allowance is earnestly solicited.

Claims 29 and 30 are allowable based on their dependency from Claim 28. Additionally, Claims 29 and 30 are allowable for based on the individual claim's own recited features which are not disclosed by Mullen. Accordingly, Applicants respectfully request that the §102(e) rejection to the instant claims be withdrawn.

Claim 32 in part recites a computer including memory configured to maintain:

- a configuration file including a second configuration setting of a second application having a second field and a second description of a second condition for the second field; and
- a configuration module that, when executed on the processor, validates the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition, and if met, then determining that the first configuration setting is valid for use with the second application.

Mullen fails to anticipate Claim 32 as Mullen does not include a configuration module as recited. Mullen paragraph [0029] is cited as teaching these features. The Office's assertion is incorrect as Mullen does not include a configuration module that, when executed . . . validates the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition. Mullen does not do this as the asserted second application is merely a copy of the first and does not require verification. In this manner, there is no need in Mullen for the asserted teachings. Removal of the pending rejection is respectfully requested.

Claims 33 and 34 are allowable based on their dependency from Claim 32. Additionally, Claims 32 and 34 are allowable for based on the individual claim's own recited features which are not disclosed by Mullen. Accordingly, Applicants respectfully request that the §102(e) rejection to the claims be withdrawn.

Claim 36 is rejected under 35 U.S.C. §102(e) over Mullen. Applicants disagree. Claim 36 has been amended. Claim 36 in part recites

- a broadcast transmitter configured to provide media content to a client in response to a request from the client;
- a first application that when executed provides media content for broadcast by the broadcast transmitter, wherein the application is composed of computer instructions, the computer instructions have attributes, and each said attribute provides a description of a condition for a field;

Mullen fails to disclose a content server which includes a broadcast transmitter or memory as recited in the claim. Mullen only discloses a system for installing program components at no time does the Mullen reference disclose providing media content as recited in Claim 36. Removal of the pending rejection is respectfully requested.

Claim 40 is allowable as nowhere does Mullen teach a "configuration module that is executable to validate whether each said condition is met by a second application." Mullen only discloses a copy functionality and does not include a configuration module that is executable to validate whether each said condition is met. The installed application of Mullen 128 does not include any comparison function. Rather, item 128 is only the copy of the first application. *See Mullen, Page 3, paragraph [0028].* Removal of the pending rejection is respectfully requested and allowance is solicited.

Claim 41 is allowable as being based on an independent claim which is believed to be in a condition for allowance. Additionally, nowhere does Mullen disclose a broadcast transmitter as recited in Claim 41.

Accordingly, it is respectfully submitted that a *prima facie* case of obviousness has not been established. Therefore, the Applicant respectfully requests that the Board overturn the First Ground of Rejection.

<u>Second Ground of Rejection</u> Claims 2-4, 10-12, 15-17, 20-22, 25-27, 31, 35, 37-39 and 42-47 satisfy the requirements of 35 U.S.C. § 103(a) and therefore are not unpatentable over Mullen in view of Joory.

Claims 2-4 are allowable based on their dependency from Claim 1. The present rejection to Claims 2-4 is improper as neither Mullen, nor Joory individually, or in combination, teach or suggest the recited features. In particular, Joory fails to correct the deficiencies in Mullen including the failure to teach validating as recited in Claim 1. The Joory reference is not cited as correcting this deficiency. Additionally, the cited passage of Joory fails to teach "wherein the first and second conditions are value constraints." The Joory passage does not include value constraints as recited. The Joory attributes disclose data field which do not necessarily teach value constraints. Exemplary value constraints including, for example, integer range, float range, or value set. Instead, the Joory attributes are merely varying data. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Ryoka*, 180 U.S.P.Q. 580 (C.C.P.A. 1974). *See als*o *In re Wilson*, 165 U.S.P.Q. 494 (C.C.P.A. 1970). Removal of the pending rejection to Claims 2-4 is respectfully requested.

Regarding **Claims 10-12**, the Joory reference fails to correct the deficiencies of Mullen including the failure to teach examining and comparing as recited in the base claim. Claims 10-12 are also allowable for their own recited features which, in combination with those recited in Claim 9, are neither shown nor suggested in the references of record, either singly or in combination with one another. Joory additionally fails to teach value constraints. Joory, Paragraph [0049] discloses various data types included within application setup data or attributes. The attributes do not teach value constraints as recited in Claim 10. The Joory attributes only provide examples of data types and do not teach or suggest the utilization of value constraints as recited. Removal of the pending rejection to Claims 10-12 is respectfully requested.

Claims 15-17 are allowable as being based on an independent claim, Claim 14, which is believed to be in a condition for allowance. The Joory reference is not cited as correcting the deficiencies of Mullen with respect to the base Claim 14. Claims 15-17 are also allowable for their own recited features which, in combination with those recited in Claim 14, are neither shown nor suggested in the references of record, either singly or in combination with one another. With particular regard to Claim 15, the Joory reference fails to teach "a value constraint selected from the group consisting of: an integer range; a float range; a value set; a string pattern; cardinality of a collection; a mandatory value; and an optional value. The recited passage of Joory fails to disclose the recited potential constraint values and does not teach each and every limitation as required for a combination under 35 U.S.C. §103(a). Removal of the pending rejection is requested and allowance is solicited.

Claims 20-22 are allowable based on their respective dependency from Claim 19 which is believed to be in a condition for allowance. The asserted combination of Mullen and Joory is improper as the Joory reference is not cited as correcting the deficiencies in the Mullen reference with respect to the features of Claim 19. Applicants will not burden the record. With specific regard to Claim 20 the Joory reference does not teach or suggest constraint values as recited in the claim. The "attributes" of Joory are simply data and do not teach or suggest "wherein each condition is a value constraint . . ." Removal of the pending rejection to the instant claims is respectfully requested.

Claims 25-27 are allowable based on their dependency from Claim 24. Claim 25 is additionally patentable as neither Joory, nor Mullen teach or disclose value constraints. Claims 26 and 27 are also allowable for their own recited features which, in combination with those recited in Claim 24, are neither shown nor suggested in the references of record, either singly or in combination with one another. Removal of the pending rejection is respectfully requested.

Claims 31 and 35 are allowable based on their dependency from Claims 28 and 32. The instant claims are also allowable for their own recited features which, in combination with those recited in each claims respective base claim, are neither shown nor suggested in the references of record, either singly or in combination with one another. Removal of the pending rejection is respectfully requested.

Claim 37 is allowable as neither the Mullen reference, nor the Joory reference teach or disclose value constraints. Joory, asserted as teaching "value constraints" instead merely teaches "attributes" which fail to disclose or suggest value constraints. Claim 37 is also allowable based on the same rationale as discussed with respect to independent Claim 36, from which the instant claim depends. Removal of the pending rejection is requested and allowance is solicited.

Claims 38 and 39 are allowable based on their dependency from Claim 36. The instant claims are also allowable for their own recited features which, in combination with those recited in each claims respective base claim, are neither shown nor suggested in the references of record, either singly or in combination with one another. Removal of the pending rejection is respectfully requested.

Claim 42 is allowable as neither the Mullen reference, nor the Joory reference teach or disclose value constraints or the features of independent Claim 40, from which the instant claim depends. Joory, asserted as teaching "value constraints" instead merely teaches "attributes" which fail to disclose or suggest value constraints. Claim 42 is also allowable based on the same rationale as discussed with respect to independent Claim 40. Removal of the pending rejection is requested and allowance is solicited.

Claims 43 and 44 are allowable based on their dependency from Claim 40. The instant claims are also allowable for their own recited features which, in combination with those recited in each claims respective base claim, are neither shown nor suggested in the references of record, either singly or in combination with one another. Removal of the pending rejection is respectfully requested.

Claim 45 is pending a rejection under 35 U.S.C. §103(a) over Mullen in view of Joory. Applicants disagree. The Mullen/Joory combination fails to teach or suggest the inclusion of a configuration module that is executable by the second computer to validate the first configuration setting. Mullen was cited for this proposition. Joory was neither cited as correcting this deficiency, nor does Joory correct this deficiency in Mullen. The Mullen reference does not validate setting because the Mullen system is only for copying. The Mullen copy functionality neither discloses validation, nor is there any need to validate as Mullen is merely concerned with computer intra copying while Claim 45 recites a first and second computer.

The citation to Joory is additionally incorrect as the combination is taken in a piecemeal fashion to cobble together the features of the instant claim rather than considering the claim as a whole, as is required under the law. Even if one were to combine Mullen and Joory, the resultant combination would not result in the present invention as there is no teaching or suggestion in the references, either singly or together, which would cause one of skill in the art to take the asserted "configuration module" and include the module in a second computer. Even if one were to assume the combination was proper, there is no teaching which would suggest modifying the

Mullen system, which is directed to a single device, and changing it to a first and second computer system with an asserted configuration module in the second computer and operate the system in the recited manner. Removal of the pending rejection is respectfully requested and allowance is solicited.

Claims 46 and 47 are allowable based on their dependency from Claim 45. The instant claims are also allowable for their own recited features which, in combination with those recited in each claims respective base claim, are neither shown nor suggested in the references of record, either singly or in combination with one another. Removal of the pending rejection is respectfully requested.

Accordingly, it is respectfully submitted that a *prima facie* case of obviousness has not been established. Therefore, the Applicant respectfully requests that the Board overturn the Second Ground of Rejection.

CONCLUSION

The Applicant respectfully considers this application to be in condition for allowance and respectfully requests the Board to overturn the final rejection and that the Examiner pass this application to allowance.

Date: March 5, 2007.

Respectfully submitted,

/William J. Breen, III #45,313/
WILLIAM J. BREEN, III
Attorney for Applicant
Registration No. 45,313

Sadler, Breen, Morasch and Colby, p.s.
422 W. Riverside Ave., Suite 424
Spokane, WA 99201
509.755.7253

APPENDIX: CLAIMS ON APPEAL

1. (Original): A method comprising validating a configuration setting of a first application for use with a second application, wherein:

the configuration setting including a first field and a first description of a first condition for the first field;

the second application is composed of computer instructions, the computer instructions having an attribute, the attribute providing a second description of a second condition for a second field; and

the validating includes:

if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition; and

if met, then determining that the configuration setting is valid for use with the second application.

2. (Previously presented): A method as described in claim 1, wherein the first and second conditions are value constraints selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

3. (Original): A method as described in claim 1, wherein the first and second conditions are default values.

4. (Original): A method as described in claim 1, wherein the first and second conditions are textual descriptions selected from the group consisting of:

a description of units in which the respective first and second fields are expressed;

a description of meanings of the respective first and second fields; and

an example of values of the first and second fields.

5. (Original): A method as described in claim 1, wherein the attribute:

is a declarative tag that that may be retrieved from and during execution of the second application; and

does not determine the value of the field.

6. (Original): A method as described in claim 1, further comprising communicating a result of the validating to the first application.

7. (Original): A method as described in claim 1, wherein the configuration setting further comprises a field type.

8. (Original): One or more computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 1.

9. (Original): A method comprising:

reading a first configuration setting of a first application including a first field and a first description of a first condition for the first field;

examining a second application to find a second configuration setting that corresponds to the first configuration setting, wherein the second application is composed of computer instructions, the computer instructions having an attribute, the attribute providing a second description of a second condition for a second field, the second configuration setting having the second field and the second description; and

comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application.

10. (Previously presented): A method as described in claim 9, wherein the first and second conditions are value constraints selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

11. (Original): A method as described in claim 9, wherein the first and second conditions are default values.

12. (Original): A method as described in claim 9, wherein the first and second conditions are textual descriptions selected from the group consisting of:

a description of units in which the respective first and second fields are expressed;

a description of meanings of the respective first and second fields; and

an example of values of the first and second fields.

13. (Original): One or more computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 9.

14. (Previously presented): A method comprising:

executing a documenter to find a plurality of fields in an application, wherein:

the application is composed of computer instructions;

the computer instructions having attributes; and

each said attribute providing a description of a condition for a respective said field,

forming a configuration file having a plurality of configuration settings of the application, wherein each said configuration setting includes one said field and the description of the condition for the one said field;

outputting the configuration file; and

utilizing the configuration file to validate configuration settings.

15. (Previously presented): A method as described in claim 14, wherein the condition is a value constraint selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

16. (Original): A method as described in claim 14, wherein the condition is a default value.

17. (Original): A method as described in claim 14, wherein the condition is a textual description selected from the group consisting of:

a description of a unit in which the respective said field is expressed;

a description of meanings of the respective said field; and

an example of a value of the respective said field.

18. (Original): One or more computer-readable media comprising computer-executable instructions that perform the method as recited in claim 14.

19. (Original): A method comprising:

generating a configuration file having a plurality of configuration settings derived from a first application, wherein:

the application is composed of computer instructions;

the computer instructions having attributes;

each said attribute providing a description of a condition for a field; and

each said configuration setting having one said field and a corresponding said description; and

validating whether the first application is valid for use with a second said application by comparing each said configuration setting of the first application with a corresponding said configuration setting of the second said application to determine whether each said condition of the first application is met by a corresponding said condition of the second application.

20. (Previously presented): A method as described in claim 19, wherein each condition is a value constraint selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

21. (Original): A method as described in claim 19, wherein each condition is a default value.

22. (Original): A method as described in claim 19, wherein each condition is a textual description selected from the group consisting of:

a unit in which a corresponding said field is expressed;

description of a meanings of a corresponding said field; and

an example of a value a corresponding said field.

23. (Original): One or more computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 19.

24. (Original): A computer-readable medium comprising computer-executable instructions that, when executed by a computer, direct the computer to:

read a first configuration setting of a first application that includes a first field and a first description of a first condition for the first field; and

validate whether the first condition is met by a second application, wherein:

the second application is composed of computer instructions;

the computer instructions have an attribute that provides a second description of a second condition for a second field; and

the first condition is validated through comparison with the second condition.

25. (Previously presented): A computer-readable medium as described in claim 24, wherein the first and second conditions are value constraints selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

26. (Original): A computer-readable medium as described in claim 24, wherein the first and second conditions are default values.

27. (Original): A computer-readable medium as described in claim 24, wherein the first and second conditions are textual descriptions selected from the group consisting of:

a description of units in which the first and second fields are expressed;

a description of meanings of the first and second fields; and

an example of values of the first and second fields.

28. (Original): A computer comprising:

a processor; and

memory configured to maintain:

a first application composed of computer instructions, the computer instructions having an attribute, the attribute providing a first description of a first condition for a first field;

a configuration file including a configuration setting of a second application having a second field and a second description of a second condition for the second field; and

a configuration module that, when executed on the processor, validates the configuration setting for use with the first application by comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met, then determining that the configuration setting is valid for use with the first application.

29. (Original): A computer as described in claim 28, wherein the second

application is stored in the memory.

30. (Original): A computer as described in claim 28, wherein the configuration file is received in a transmission from a network for storage in the memory.

31. (Original): A computer as described in claim 28, wherein the first and second conditions are selected from the group consisting of:

a value constraint;

a default value;

a description of units in which the first and second fields are expressed;

a description of meanings of the first and second fields; and

an example of values of the first and second fields.

.

32. (Original): A computer comprising:

a processor; and

memory configured to maintain:

a first application composed of computer instructions, the computer instructions having an attribute, the attribute providing a first description of a first condition for a first field, and wherein a first configuration setting includes the first description and the first field;

a configuration file including a second configuration setting of a second application having a second field and a second description of a second condition for the second field; and

a configuration module that, when executed on the processor, validates the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition, and if met, then determining that the first configuration setting is valid for use with the second application.

33. (Original): A computer as described in claim 32, wherein the second application is stored in the memory.

34. (Original): A computer as described in claim 32, wherein the configuration file is received over a network and stored in the memory.

35. (Original): A computer as described in claim 32, wherein the first and second conditions are selected from the group consisting of:

a value constraint;

a default value;

a description of units in which the first and second fields are expressed;

a description of meanings of the first and second fields; and

an example of values of the first and second fields.

36. (Previously presented): A content server comprising:

a broadcast transmitter configured to provide media content to a client in response to a request from the client;

a processor; and

memory configured to maintain:

a first application that when executed provides media content for broadcast by the broadcast transmitter, wherein the application is composed of computer instructions, the computer instructions have attributes, and each said attribute provides a description of a condition for a field; and

a documenter that is executable on the processor to generate a configuration file having a configuration setting of the first application, wherein the configuration setting includes the field and the description of the condition for the field.

37. (Previously presented): A content server as described in claim 36, wherein the condition is a value constraint selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

38. (Previously presented): A content server as described in claim 36, wherein the condition is a default value.

39. (Previously presented): A content server as described in claim 36, wherein the condition is a textual descriptions selected from the group consisting of:

a description of units in which the first and second fields are expressed;

a description of meanings of the first and second fields; and

an example of values of the first and second fields.

40. (Original): A content server comprising:

a first application composed of computer instructions, the computer instructions having attributes, and each said attribute providing a description of a condition for a field;

a configuration module that is executable to validate whether each said condition is met by a second application; and

a documenter that is executable to generate a configuration file having a configuration setting of the first application, wherein the configuration setting includes the field and the description of the condition.

41. (Original): A content server as described in claim 40, further comprising a broadcast transmitter, wherein the first application, when executed, provides content for broadcast by the broadcast transmitter.

42. (Previously presented): A content server as described in claim 40, wherein each said condition is a value constraint selected from the group consisting of:

an integer range;

a float range;

a value set;

a string pattern;

cardinality of a collection;

a mandatory value; and

an optional value.

43. (Original): A content server as described in claim 40, wherein each said

condition is a default value.

44. (Original): A content server as described in claim 40, wherein each said condition is a textual description selected from the group consisting of:

a description of a unit in which a respective said field is expressed;

a description of a meaning of a respective said field; and

an example of a values of a respective said field.

45. (Original): A system comprising:

a network;

a first computer communicatively coupled to the network and including a first application composed of computer instructions, the computer instructions having a first attribute that provides a first description of a first condition for the first field, wherein a first configuration setting includes the first field and the first description;

a second computer communicatively coupled to the network and including:

a second application composed of computer instructions having a second attribute that provides a second description of a second condition for a second field; and

a configuration module that is executable by the second computer to validate the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition.

46. (Original): A system as described in claim 45, wherein the first computer is configured as a set-top box and the second computer is configured as a content server.

47. (Original): A system as described in claim 45, wherein the first and second conditions are selected from the group consisting of:

a value constraint;

a default value;

a description of units in which the first and second fields are expressed;

a description of meanings of the first and second fields; and

an example of values of the first and second fields.

APPENDIX: EVIDENCE

None.

APPENDIX: RELATED PROCEEDINGS

None.